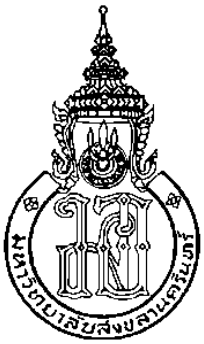


รายงานการทดลองที่ 1

เรื่อง Introduction to Python

เสนอ

อาจารย์ ฐิตินันท์ เกลี้ยงสุวรรณ

จัดทำโดย

ชื่อ นาย ปณิธาน ดวงขวัญ รหัสนักศึกษา 5735512036

ภาคการศึกษาที่ 1 ปีการศึกษา 2560

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

การทดลองที่ 3SA01

เรื่อง Introduction to Python

วัตถุประสงค์

1. รู้จักวิธีพัฒนาโปรแกรมด้วยภาษา Python3
2. เข้าใจการใช้งานโครงสร้างข้อมูลลิสต์ ดิคชันนารี และโครงสร้างควบคุมเบื้องต้น
3. ศึกษาการทำงานฟังก์ชั่นและไลบรารี่ต่างๆ จากเอกสารที่มาพร้อมกับตัวติดตั้ง Python 3 พร้อมทั้งเรียกใช้งานฟังก์ชั่นได้อย่างเหมาะสม

การทดลองที่ 1.1

1.1 NUMBERS

```
>>> a = 7
>>> b = a + 10
>>> b
```

*หากใช้ python ใน interactive mode เราสามารถโค้ดเพียงแค่นิพจน์ที่ต้องการทราบค่า ตัว python interpreter จะแสดงค่าของนิพจน์ให้โดยอัตโนมัติ

```
>>> import math
>>> a = 3/2
>>> a
>>> math.ceil(a)
```

*math เป็น module หนึ่งของ python ที่สามารถใช้งานได้ เมื่อทำการ import

```
math.ceil(x)

Return the ceiling of x, the smallest integer greater than or equal to x. If x
is not a float, delegates to x.__ceil__(), which should return an Integral
value.
```

CP-01

กำหนดตัวแปร x ให้มีค่าเป็น `0.3` จงหาค่าของ `1.5*sin(x)`

รูปภาพประกอบ Checkpoint

```
>>> x = 0.3
>>> z = 1.5 * sin(x)
>>> z
0.4432803099920093
>>> ceil(x)
1
```

สรุปผลการทดลองที่ 1.1

จากรูปภพข้างต้นจะแสดงให้เห็นว่าหลังจากเรา import ฟังก์ชั่น math แล้วเราสามารถเรียกใช้ฟังก์ชั่นได้เลยโดยที่ใช้จะเป็นฟังก์ชั่น sin cos และ tan โดยสามารถปัดเลขทศนิยมได้จากฟังก์ชั่น ceil

การทดลองที่ 1.2

1.2 STRINGS

สังเกตความแตกต่างของการใช้ single quote, double quote และ triple quote

```
>>> 'spam eggs'
>>> 'doesn\'t'
>>> "doesn't"
>>> '"Yes," he said.'
>>> "\"Yes,\" he said."
>>> '"Isn\'t," she said.'
>>> print("""\
Usage: thingy [OPTIONS]
     -h                        Display this usage message
     -H hostname               Hostname to connect to
""")
```

วิธีการใช้หมายเลข index แสดงข้างล่าง

```
 +---+---+---+---+---+
 | H | e | l | p | A |
 +---+---+---+---+---+
 0   1   2   3   4   5
-5  -4  -3  -2  -1
```

สังเกตการใช้งาน [ ] เพื่อหาค่าของตัวอักษร ณ. ตำแหน่งที่ต้องการ รวมถึงการกำหนดช่วงของตัวอักษร

```
>>> line = 'hello world'
>>> line[4]
>>> line[0:2]
>>> line[2:4]
>>> line[:4]
>>> line[4:]
>>> line[1:100]
>>> line[-1]
>>> line[-5:-1]
```

สังเกตการใช้งานฟังก์ชันเกี่ยวกับ String

```
>>> line = 'hello mars'
>>> len(line)
>>> line.count('l')
```

เรียนรู้การสร้างข้อความตามรูปแบบที่กำหนด

```
>>> x = 10/3
>>> "The value of x is {0}.".format(x)
>>> "The value of x is {0:.3f}.".format(x)
>>> 'Coordinates: {lat}, {lon}'.format(lat='37.24N', lon='-115.81W')
```

CP-02

จงดำเนินการกับข้อความ “Once upon the time” ดังต่อไปนี้

- สร้างข้อความใหม่ จากข้อความเดิมตั้งแต่ตำแหน่ง 3 ถึง 9
- แสดงข้อความเดิม ตามด้วยข้อความใหม่ โดยมีเครื่องหมาย => กั้นระหว่างข้อความ โดยใช้ฟังก์ชัน `format()` อย่างเหมาะสม

รูปภาพประกอบ Checkpoint

```
>>> b = "e upon"
>>> line + " => {}".format(b)
```

สรุปผลการทดลองที่ 1.2

จากการทดลองจะพบว่าเราสามารถแยกส่วนของลิสต์ได้โดยจะแยกด้วยคำสั่งของมันเองโดยจะใช้ index ในการแยกออกมาหลังจากนั้นเราสามารถเก็บค่าไว้ในตัวแปรใดตัวแปรหนึ่ง โดยเราสามารถปริ้นข้อความเก่าบวกกับข้อความใหม่ได้โดยใช้คำสั่ง line + “….” โดยจะมีฟังก์ชั่น format อ้างอิงภายใต้เครื่องหมายดับเบิ้ลโค้ด

การทดลองที่ 1.3

1.3 LISTS

การใช้งานพื้นฐาน List ของ Python

```
>>> a = ['spam', 'eggs', 100, 1234]
>>> a[0]
>>> a[-2]
>>> a[1:-1]
>>> a[:]
```

การจัดการข้อมูลใน List

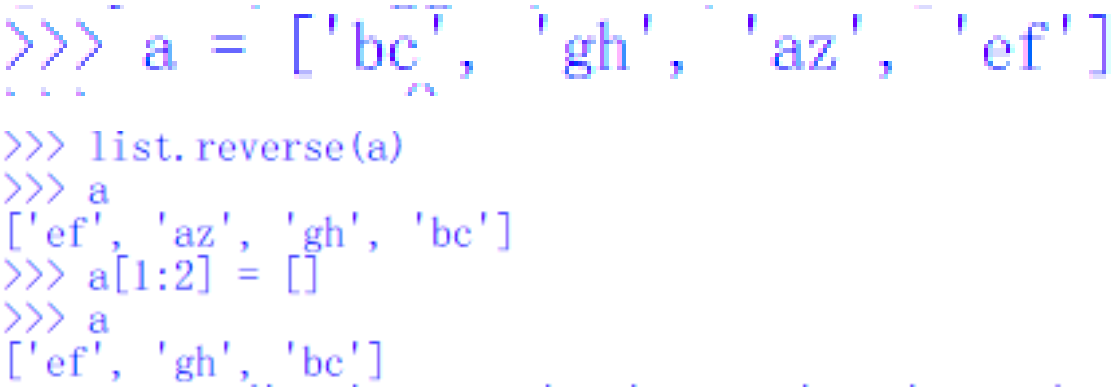
```
>>> a = ['spam', 'eggs', 100, 1234]
>>> a[2] = a[2] + 23
>>> a
>>> a[0:2] = [1, 12]
>>> a
>>> a[0:2] = []
>>> a
>>> len(a)
```

CP-03

กำหนดให้ลิสต์ a มีค่าดังนี้ `['bc', 'gh', 'az', 'ef']` จงดำเนินการกับลิสต์ a ดังต่อไปนี้

- สั่ง reverse สมาชิกในลิสต์ โดยใช้ฟังก์ชัน `reverse()`
- ลบสมาชิกหมายเลข 2
- แสดงจำนวนสมาชิกในลิสต์

รูปภาพประกอบ Checkpoint



สรุปผลการทดลองที่ 1.3

จากภาพประกอบจะเห็นได้ว่า หลังเราประกาศลิสต์ออกมาแล้วเราจะใช้คำสั่ง reverse โดยคำสั่งนี้จะทำหน้ากลับด้านเรียงจากหลังมาหน้าแทนและหลังจากนั้นเราจะให้ลิสต์ช่องที่สองหายไปโดยการกำหนด index ของลิสต์ที่สองให้เท่ากับช่องว่างหรือจะเขียนคำสั่ง remove ลงไปก็ได้และหลังจากนั้นก็แสดงลิสต์ a ออกมาใหม่

การทดลองที่ 1.4

1.4 DICTIONARIES

การใช้งานพื้นฐานดิคชันนารีของ Python

```
>>> tel = {'jack': 4098, 'sape': 4139}
>>> tel['guido'] = 4127
>>> tel
>>> tel['jack']
>>> del tel['sape']
>>> tel['irv'] = 4127
>>> tel
>>> list(tel.keys())
>>> sorted(tel.keys())
>>> 'guido' in tel
>>> 'jack' not in tel
```

CP-04

กำหนดให้ดิคชันนารี dict มีค่าดังนี้ {'dog' : 100, 'cat' : 230, 'bird' : 120} จงดำเนินการกับดิคชันนารี dict ดังต่อไปนี้

- เพิ่มค่าของ dog อีก 70
- ทำการลบสมาชิก bird

รูปภาพประกอบ Checkpoint

```
>>> dict = {'dog' : 100, 'cat': 230, 'bird' : 120}
>>> dict
{'dog': 100, 'cat': 230, 'bird': 120}
>>> dict.update({'dog' : 170})
>>> dict
{'dog': 170, 'cat': 230, 'bird': 120}

>>> dict.pop('bird')
120
>>> dict
{'dog': 170, 'cat': 230}
```

สรุปผลการทดลองที่ 1.4

จากการทดลองจะเห็นได้ว่าในตัวของภาษานั้นเราสามารถทำเป็นดิคชันนารี่ได้โดย ตัวแปรหรือคำแต่ละคำนั้นจะมีค่าของคำมันอยู่ในตัวจากภาพจะเห็นว่าเรากำหนดคำว่า dog ให้มีค่าเท่ากับ 100 เราสามารถหาคำว่า dog ได้จากตัวเลขได้นั่นเอง หลังจากนั้นเราสามารถอัพเดทค่าต่างๆที่เราต้องการลงไปในดิคชันนารี่ได้โดยการใช้ฟังก์ชั่นอัพเดทและนอกจากนั้นเราสามารถนำข้อมูลออกจากดิกชันนารี่ได้โดยใช้คำสั่ง pop

การทดลองที่ 1.5

1.5 FLOW CONTROLS

การใช้งาน `if/else` โดยมีข้อสังเกตุว่า Python ใช้การ indent แทนการกำหนดขอบเขตโดย { และ }

```
>>> x = int(input("Enter: "))
>>> if x < 0:
    print('Negative')
elif x == 0:
   print('Zero')
else:
    print('Positive')
```

การใช้งาน `for` เพื่อเข้าถึงสมาชิกในลิสต์

```
>>> a = ['cat', 'window', 'defenestrate']
>>> for x in a:
    print(x, len(x))
```

การใช้งาน `for` ร่วมกับฟังก์ชัน `range`

```
>>> for i in range(5):
     print(i)

>>> for i in range(3, 7):
     print(i)
```

CP-05

วนลูปเพื่อรับตัวเลขจำนวน 5 ตัว ใส่ไว้ในลิสต์ หลังจากนั้นหาค่าผลรวมของสมาชิกในลิสต์ โดยประมวลผลเฉพาะตัวเลขที่มีค่าเป็นบวกเท่านั้น

รูปภาพประกอบ Checkpoint

```
>>> for i in range(5):
...     x[i] = int(input("Enter: "))
...
Enter: 5
Enter: -3
Enter: 5
Enter: -2
Enter: 5
```

```
>>> for i in range(5):
...     if(x[i] > 0):
...             sum = sum + x[i]
...
>>> sum
15
```

สรุปผลการทดลองที่ 1.5 checkpoint 05

จากภาพประกอบด้านบนจะพบว่าเราสามารถประมวลผลเป็นอาร์เรย์ได้เหมือนกับภาษาซีโดยจะรับค่าเก็บไว้ในลิสต์ Xโดยใช้คำสั่งวนลูป for โดยจะมีการวนลูปทั้งหมด 5 ครั้งโดยจะเก็บค่าไว้ในตัวแปล X และหลังจากนั้นเราจะวนลูปอีกครั้งโดยเก็บค่าไว้ใน sum โดยการนำค่าที่อยู่ใน ลิสต์ X นั้นจะมาบวกกันทุกค่าและเก็บค่าที่บวกไว้ในตัวแปร sum และแสดงผลออกมา

CP-06

รับข้อความหนึ่งบรรทัด แล้วแบ่งออกเป็นคำๆ ใส่ไว้ในลิสต์ (ใช้ฟังก์ชัน `split`) วนลูปเพื่อลบคำที่มีความยาวน้อยกว่า 3 ตัวอักษร และแสดงผลคำตามลำดับ (ใช้ฟังก์ชัน `sorted`) โดยไม่เปลี่ยนแปลงลำดับในอาเรย์เดิม

```
>>> s = input("Enter sentence: ")
Enter sentence: This is PSU
>>> s
'This is PSU'
>>> b = s.split()
>>> b
['This', 'is', 'PSU']
>>> for i in range(3):
...     if(len(b[i]) < 3):
...             b.remove(i)
...
>>> for i in range(3):
...     if(len(b[i]) < 3):
...             del b[i]
...
>>> b
['This', 'PSU']
>>> sorted(b)
['PSU', 'This']
```

สรุปผลการทดลองที่ 1.5 Checkpoint 06

จากภาพประกอบด้านบนจะพบว่าเรา input ข้อมูลเป็นประโยคมา 1 ประโยคแล้วหลังจากนั้นเราใช้ฟังก์ชั่น split() เพื่อทำหน้าที่ในการแยกคำออกโดยเก็บไว้ในตัวแปร b และตัวแปร b จะเป็นตัวแปรที่เป็นลิสต์โดยจะแยกไว้ 3 คำจากภาพเนบน หลังจากนั้นเราใช้คำสั่งวนลูปในการหาคำที่มาความยาวตัวอักษรน้อยกว่าสามตัวโดยถ้าเราเจอคำนั้นเราจะใช้คำสั่งลบออกจากลิสต์ทันทีและแสดงตัวแปร b หลังจากลบเสร็จเรียบร้อย สุดท้ายใช้คำสั่ง sorted() ในการเรียงลำดับคำโดยจะเรียงลำดับตัวอักษร

การทดลองที่ 1.6

1.6 GENERATOR EXPRESSIONS

Generator Expression มักถูกใช้กับการประมวลผลคอลเล็คชัน เช่นลิสต์ เพื่อความสะดวกในการเขียนโคด

```
>>> a = [2, 4, 7, 8]
>>> sum(a)
>>> sum(x*x for x in a)              //generator is bold.
```

```
>>> b = {'dog':10, 'cat':7, 'bat':2}
>>> v, k = max((v,k) for k,v in b.items())
>>> "{0} has max value at {1}".format(k, v)
```

```
>>> a = [3, 2, 1, -7]
>>> sum(x for x in a if x > 0)
>>> max(abs(x) for x in a)
```

CP-07

รับข้อความหนึ่งบรรทัด ใช้ประโยชน์จาก Generator Expression เพื่อหาคำที่มีความยาวสูงสุด

รูปภาระประกอบ Checkpoint

```
>>> s = input("Enter sentence: ")
Enter sentence: I LOVE PSU
>>> m = s.split()
>>> max(len(x) for x in m)
4
```

สรุปผลการทดลองที่ 1.6

จากภาพด้านบนเราจะรับประโยคมา 1 ประโยคและเก็บค่าไว้ในตัวแปร s หลังจากนั้นเราจะแยกคำออกมาเพื่อหาว่าคำไหนมีความยาวมากที่สุดโดยใช้ คำสั่ง max(len(x) for x in m) โดยคำสั่งนี้จะไปหาจำนวน len ของแต่ละคำและจะวนลูปครบเท่ากับจำนวนคำที่มีอยู่ในลิสต์ m

การทดลองที่ 2.1

2.1 FUNCTIONS

การนิยามฟังก์ชัน และการเรียกใช้งาน

```
>>> def add(a, b):
    return a + b

>>> add(3, 8)
```

ในบางโอกาส การนิยามฟังก์ชันสามารถกำหนดค่าปริยาย ทำให้การเรียกใช้ฟังก์ชันไม่จำเป็นต้องระบุพารามิเตอร์

```
>>> def f(a, L=None):
    if L is None:
        L = []
    L.append(a)
    return L

>>> f(2, f(4, f(3)))
```

CP-08

นิยามฟังก์ชัน `name` รับพารามิเตอร์ 3 ตัวคือ ชื่อ (`first`), นามสกุล (`last`) และ คำนำหน้า (`title`) โดยฟังก์ชันคืนค่าเป็นข้อความประกอบด้วย คำหน้าหน้า ชื่อ นามสกุล ตามลำดับ โดยหากไม่ได้กำหนด `title` ให้ใช้คำว่า `Khun` ยกตัวอย่างเช่น

```
name('John', 'Doe', 'Mr.')  => 'Mr. John Doe'
name('Somchai', 'Dang') => 'Khun Somchai Dang'
```

รูปภาพประกอบ Checkpoint

```
>>> def name(first,last,title=None):
...     if title is None:
...             print("khun" + ' ' + first +' ' + last)
...     else:
...             print(title + ' ' + first +' ' + last)
...
>>> name('Panitarn','Duangkhwan','Mr.')
Mr. Panitarn Duangkhwan
>>> name('Panitarn','Duangkhwan')
khun Panitarn Duangkhwan
>>>
```

สรุปผลการทดลอง

จากภาพประกอบด้านบนเราเขียนฟังก์ชั่นขึ้นมาโดยจะรับค่ามาเป็น ประโยคโดยจะเป็นชื่อนามสกุล และคำนำหน้าโดยมีเงื่อนไขว่าถ้าไม่ใส่คำนำหน้ามาจะเข้าเงื่อนไขแรกแต่ถ้าใส่มาจะเข้าเงื่อนไขสองหลังจากนั้นลองเรียกใช้ฟังก์ชั่นก็จะได้ผลตามรูปภาพประกอบด้านบน

การทดลองที่ 2.2

2.2 INPUT & OUTPUT

เขียนไฟล์

```
>>> f = open('d:/test.txt', 'w')
>>> f.write("AB\n")
>>> f.write("ZD\n")
>>> f.write("GH\n")
>>> f.close()
```

อ่านไฟล์

```
>>> f = open('d:/test.txt')
>>> for line in f:
    print(line, end='')
>>> f.close()
```

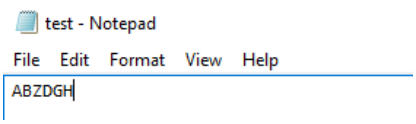
CP-09

อ่านไฟล์ `test.txt` (ที่เขียนขึ้นจากการทดลองในตอนนี้) แต่ละบรรทัดให้ตัด `'\n'` ทิ้งโดยใช้ฟังก์ชัน `strip()` แล้วนำไปใส่ไว้ในลิสต์ หลังจากสั่งเรียงลำดับข้อมูล แล้วบันทึกข้อมูลลงไปในไฟล์ในลักษณะเดิม

รูปภาพประกอบ Checkpoint

```
>>> f.write("".join(n))
0
>>> f.close()
>>> f = open('d:/test.txt', 'w')
>>> f.write("AB\n")
3
>>> f.write("ZD\n")
3
>>> f.write("GH\n")
3
>>> f.close()
>>> f = open('d:/test.txt')
>>> n=[]
>>> i=0
>>> for line in f:
        n.insert(i,line.strip('\n'))
        i+=1


>>> f.close()
>>> f = open('d:/test.txt', 'w')
>>> f.write("".join(n))
6
>>> f.close()
```



สรุปผลการทดลองที่ 2.2

จากภาพการทดลองจะพบว่าหลังจากเราใส่ข้อมูลลงไปแล้วเราจะพบว่า การเขียนข้อมูล จะลงไปอยู่ในในไฟล์ .txt และหลังจากนั้นพอเราแก้โดยใช้คำสั่ง strip ทำหน้าที่ลบ \n ออกและเราก็เซฟไฟล์เราจะพบว่าจากที่มี \n ในตอนแรกนั้น \n จะหายไปทันทีและมันจะมาอยู่ในบรรทัดเดียวกันทั้งหมด

การทดลองที่ 2.3

2.3 SOME LIBRARIES

OS & File Utilities

```
>>> import os
>>> from os.path import getsize, join
>>> os.name
>>> os.listdir("d:/lab")
>>> for root, dirs, files in os.walk('D:/lab/'):
        print(root, '=', sum(getsize(join(root,name)) for name in files))
```

```
>>> import glob
>>> glob.glob("d:/lab/*.py")
```

Regular Expression

```
>>> import re
>>> re.split('\W+', "Hello, world")

>>> r = re.compile('\W+')                    //good for reused!
>>> re.split(r, "Hello, world")

>>> m = re.match(r"(\d+)\.(\d+)", "24.1632")
>>> m.groups()
```

Web Client

```
>>> from urllib import request
>>> f = request.urlopen('http://fivedots.coe.psu.ac.th')
>>> print(f.read().decode('utf-8'))
```

CP-10

อ่านข้อมูลจากเว็บ http://fivedots.coe.psu.ac.th นำข้อมูลบรรทัดที่ 32-77 ใส่ไว้ในลิสต์ (แต่ละอีลิเมนต์คือข้อมูลแต่ละบรรทัด) ประมวลผลข้อมูลในลิสต์ เพื่อแสดงรายชื่อของอาจารย์และเจ้าหน้าที่ของภาควิชา (แสดงเฉพาะชื่อกับนามสกุลเท่านั้น) โดยให้ใช้ประโยชน์จาก Regular Expression อย่างเหมาะสม

รูปภาพประกอบ Checkpoint

```
>>> from urllib import request
>>> import re
>>> f = request.urlopen('http://fivedots.coe.psu.ac.th')
>>> x = f.read( ) . decode('utf-8') . split('\n')

>>> for i in range(32,78):
        cont = re . findall('<.+>(.+)<.+>' ,x[i])
        if (len(cont)==0):
                cont = re . findall('<.+>(.+)- ->' ,x[i])
        if (len(cont)==0):
                cont = re . findall('<.+>(.+)' ,x[i])
        cont
```

สรุปผลการทดลองที่ 2.3

จากการทดลองจะเห็นว่าเราสามารถเรียกไลบรารี่จากเว็บไซต์ข้างนอกได้นั่นเองโดยเราจะส่งค่าเพียงค่าบรรทัดที่ 32-77 เท่านั้นโดยหลังจากวนแล้วเราจะให้แสดงโดยใช้เงื่อนไขในการเช็คข้อมูลและแสดงผลออกมาจากโจทย์ที่ให้มาโดยให้แสดงชื่อ และ นามสกุลของอาจารย์และเจ้าหน้าที่เท่านั้น

รูปหลังจากเรียกแสดงผล

```
["fivedots's user homepages"]
[]
['Anan Nilagozee']
['Dr. Anant Choksuriwong']
['Dr. Andrew Davison']
['Anucha Rattana']
['Dr. Aree Teeraparbseree']
['Bongkot Prukepong']
['Chatchai Jantaraprim']
['Damrong Kalawdee']
['Darunee Suttiwipakorn']
['Grit Khaorapapong -->']
['Dr. Nikom Suvonvorn']
['Dr. Montri Karnchanadecha']
['Mallika Unhawiwat']
['Noppadon Kamolvilassatien']
['Nuanwan Ponpong']
['Paijit Kochakornjarupong']
['Paiboon Bontawin']
['Panyayot Chaikarn']
['Patimakorn Jantaraprim']
['Petcharat Suriyachai']
['Phatcharee Thepnimit']
['Dr. Pichaya Tandayya']
['Dr. Richard N. Zobel -->']
['Robert Elz']
['Sakuna Charoenpanyasak']
['Dr. Sangsuree Vasuponga yya']
['Seksun Suwanmanee']
['Dr. Sinchai Kamolphiwong']
['Somchai Limsiroratana']
['Suntichai Chuaywong -->']
['Dr. Suntorn Witosurapot']
['Suthon Sae-Whong']
['Suwat A. -->']
['Dr. Taweesak Reungpeerakul']
['Thammaratt Samitalampa']
['Dr. Thanate Khaorapapong']
['Thossaporn Kamolphiwong']
['Touchai Angchuan']
['Dr. Wannarat Suntiamorntut']
['Warodom Weerapan']
['Watcharin Kaewapichai']
['Weerapant Musigasarn -->']
['Worraprot Chukumnird']
['Jaruwat Somboon']
```

แบบฝึกหัด (Exercise)

แบบฝึกหัดข้อที่ 1

ข้อ 1 ไฟล์ employee.txt เก็บข้อมูลเกี่ยวกับพนักงานไว้ดังแสดงข้างล่าง

```
John Doe:28000
Clark Kent:27000
Jane Doe:26000
```

เขียนโปรแกรมเพื่ออ่านไฟล์ แสดงผลเงินเดือนเรียงตามเงินเดือนจากมากไปหาน้อย พร้อมทั้งผลรวมของเงินเดือน

รูปภาพประกอบแบบฝึกหัด

```
>>> f=open('d:/employee.txt')
>>> show=[]
>>> for i in f:
        a=i.strip('\n')
        b=a.split(':')
        show.append(int(b[1]))


>>> show
[28000, 27000, 26000]
>>> sorted(show)
[26000, 27000, 28000]
>>> result=sum(show)
>>> result
81000
>>>
```

คำอธิบายโค้ด : จากโปรแกรมข้างต้นจะสังเกตได้ว่าเราเรียกเปิดไฟล์ employee มาจากไดเรกทอรี่ d หลังจากนั้นประกาศตัวแปรขึ้นมาเพื่อรองรับให้เก็บค่าไว้ในนั้นโดยจะใส่ค่าไว้ในตัวแปร show และโดยค่าที่เก็บนั้นจะมาจากตัวเลขหรืออะไรก็ตามที่อยู่หลัง (:) จากคำสั่ง split หลังจากนั้นก็ให้แสดงค่า Show โดยค่า show นั้นได้เรียงลำดับจากมากไปน้อยอยู่แล้วนั่นเอง หลังจากนั้นประกาศตัวแปรขึ้นมา1ตัวแปรเพื่อรองรับค่าที่คำนวณจากตัวแปร Show ทั้งหมดและแสดงผลออกมา

แบบฝึกหัดข้อที่ 2

ข้อ 2 ไฟล์ student.txt เก็บรหัสนักศึกษาของนักศึกษา

```
4010118=g1
4110112=g2
4110222=g1
```

เขียนโปรแกรมเพื่อสร้างไดเร็คทอรีตามชื่อรหัสของนักศึกษา ภายใต้ไดเร็คทอรีนั้นๆ ให้มีไฟล์ชื่อ work.conf โดยไฟล์นี้ของนักศึกษารหัส 4110118 จะมีข้อความดังแสดงข้างล่าง

```
id=4010118
group=g1
```

รูปภาพประกอบแบบฝึกหัด

```
>>> import os
>>> f = open('d:/student.txt')
>>> line=[]
```

```
>>> for i in f:
        a=i.strip('\n')
        b=a.split('=')
        os.makedirs('d:/{0}'.format(b[0]))
        f2=open('d:/{0}work.conf'.format(b[0]),'w')
        f2.write("id = {0}\n".format(b[0]))
        f2.write("group = {0}\n".format(b[1]))


13
11
13
11
>>>
```

คำอธิบายโค้ด : จากการทดลองเราจะ import os เข้ามาเพื่อใช้งานฟังก์ชั่นต่างๆ โดยเราจะเปิดไฟล์ student.txt ที่เราสร้างไว้ในคอมพิวเตอร์ของเราหลังจากนั้นประกาศตัวแปรขึ้นมา1ตัว สร้างวนลูปโดยจะให้เก็บค่าข้อมูลที่ต้องการ โดยดูจากคำสั่ง a และ b ต่อมาสร้าง directory ขึ้นมาใหม่และทำการสร้างไฟล์ชื่อ work.conf โดยทำการเขียนข้อมูลที่เราใส่ไว้ใน b นั่นเองโดยไฟล์นั้นจะเขียนข้อมูลจากตัวแปร b ลงไป

งานท้ายการทดลอง

งานหลังการทดลอง

นศ. เขียนสคริป หรือโปรแกรมด้วย Python จำนวน 1 โปรแกรม เพื่อทำงานตามที่ นศ. ต้องการ โดยจะต้องมีการใช้ความรู้ต่างๆ ในแลบ ไม่น้อยกว่า 5 หัวข้อ (นับที่หัวข้อย่อย เช่น 1.1, 1.4, 2.1, ... เป็นต้น) โดยให้ส่งไฟล์ที่บีบอัดแล้ว เพียงไฟล์เดียวในรูปแบบ zip, rar, tar.gz หรือ 7z ซึ่งประกอบด้วย

- ไฟล์ซอร์สโค้ด
- ไฟล์ PDF บอกรายละเอียดดังต่อไปนี้
  - วัตถุประสงค์ของสคริป หรือโปรแกรม
  - องค์ความรู้จากแลบ ที่ได้นำไปใช้ในโปรแกรม
  - ผลการทำงาน
- ไฟล์อื่นๆ หากสคริป หรือโปรแกรม อ่านข้อมูลจากไฟล์

รูปภาพประกอบงานท้ายการทดลอง

```
>>> def add(x, y):
    return x + y

>>> def subtract(x, y):
    return x - y

>>> def divide(x, y):
    return x / y

>>> def multiply(x, y):
    return x * y
```

```
Select operation.
>>> add
<function add at 0x041EA4B0>
>>> print("1.Add")
1.Add
>>> print("2.Subtract")
2.Subtract
>>> print("3.Multiply")
3.Multiply
>>> print("4.Divide")
4.Divide
>>> choice = input("Enter choice(1/2/3/4):")
Enter choice(1/2/3/4):1
>>> num1 = int(input("Enter first number: "))
Enter first number: 5
>>> num2 = int(input("Enter second number: "))
Enter second number: 3
```

```
>>> if choice == '1':
        print(num1,"+",num2,"=", add(num1,num2))
elif choice == '2':
        print(num1,"-",num2,"=", subtract(num1,num2))
elif choice == '3':
        print(num1,"*",num2,"=", multiply(num1,num2))
elif choice == '4':
        print(num1,"/",num2,"=", divide(num1,num2))
else:
        print("Invalid input")


5 + 3 = 8
```

คำอธิบายโค้ด : จากโปรแกรมข้างต้นจะเป็นการทำงานของฟังก์ชั่นต่างๆโดยจะทำเป็นโปรแกรมเครื่องคิดเลขโดยสามารถดูได้จากโค้ดโดยขั้นแรกจะสร้างฟังก์ชั่นต่างๆของเครื่องคิดเลข คือ การบวก ลบ คูณ หาร มีทั้งหมด4 ฟังก์ชั่นหลังจากนั้นจะมีการให้เลือกว่าเราต้องการใช้ฟังก์ชั่นไหนต่อมาจะให้ใส่ค่าตัวเลข2ค่าโดยตัวเลขทั้งสองค่าจะเก็บไว้ในตัวแปร num1 และ num2 พอเก็บเสร็จแล้วทั้งสองค่าที่เก็บไว้นั้นจะเข้าสู้การดำเนินการเงื่อนไขโดยเงื่อนไขมาจากถ้าเราเลือก operation แบบไหนมันก็จะเข้า operation แบบนั้นเองจากรูปเราเลือก choice 1 หมายความว่าเราเลือกเครื่องหมายบวกต่อมาจะเข้าในเงื่อนไข choice มีค่าเท่ากับ 1 คือจะทำการแสดงผลข้อมุลโดยการใช้ฟังก์ชั่น add นั่นเองโดยในฟังก์ชั่นก็จะคืนค่า num1 + num2 นั่นเอง